Rajesh Sawant

Tempe, AZ | 480-685-0404 | rsawant5@asu.edu | linkedin.com/in/rajesh-sawant11 | github.com/rajeshsawant98

EDUCATION

Master of Science in Computer Software Engineering — Expected May 2026
Arizona State University, Tempe, AZ — *GPA 4/4*

Coursework: Data Structures and Algorithms, Object Oriented Programming, Cloud Computing, Data Processing at Scale, Big Data, Data Mining, Semantic Web, Software Security and Testing, Software Project Management etc.

TECHNICAL SKILLS

Programming Languages: Java, Python, C/C++, SQL, JavaScript, HTML/CSS, MATLAB
Frameworks and Libraries: ReactJS, FastAPI, Node.js, Bootstrap, JUnit, Spring Boot, Tailwind, Material-UI
Tools and Technologies: Visual Studio Code, Mendix Studio Pro, Eclipse, Postman, Figma, AWS Lambda, GitHub
Databases: PostgreSQL, Firestore, MySQL, MongoDB, NoSQL, GraphDB, DynamoDB
Cloud: Google Cloud Platform (GCP), Amazon Web Services AWS (EC2, S3), Microsoft Azure, Firebase, IAM
DevOps & Project Management: Git, GitHub Actions, SVN, Docker, Kubernetes, Terraform, Jira, Confluence

WORK EXPERIENCE

Software Engineer (Full-Stack Application Developer) — August 2021 – July 2024
Intelizign Lifecycle Services — *Pune, India*

- Contributed to the development and deployment of full-stack web and native applications for clients in automotive, manufacturing, and PLM domains, focusing on optimizing performance, scalability, and user experience.
- Engineered secure authentication mechanisms using OAuth 2.0, Azure AD, and OIDC SSO, ensuring compliance with security protocols and safeguarding enterprise data.
- Deployed Mendix applications in BMW's AWS (EC2, EKS) cloud infrastructure using Kubernetes, Docker and Terraform optimizing scalability, performance, and security for high-traffic applications.
- Automated data synchronization and system integrations between Mendix and Teamcenter using TcConnector, RESTful APIs, and SOA services, streamlining workflows for data synchronization, reporting, and lifecycle management, which improved operational efficiency by 30%.
- Implemented ETL pipelines to process large-scale product data from multiple sources into unified databases, ensuring data integrity and efficient transformation for reporting and analytics.
- Developed a recruitment automation system featuring resume parsing, referral tracking, and interview scheduling with Microsoft Graph APIs and Azure AD.
- Designed and developed scalable, reusable, and mobile-responsive UI components using ReactJS, Mendix, and SCSS, focusing on modern, intuitive, and user-friendly experiences.
- Collaborated with cross-functional Agile teams to deliver scalable solutions, participating in sprint planning, code reviews, and CI/CD pipeline optimizations.

PROJECTS

Travel Itinerary Planner using Semantic Web | *SPARQL, FastAPI, React.js* — October 2024 – December 2024

- Designed an intelligent travel planner web application using semantic web principles, generating travel itineraries based on user profiles and preferences.
- Performed ETL on multiple datasets using Ontotext Refine for GraphDB integration. Defined SPARQL query templates for GeoFeature identification based on user preferences and distance constraints.
- Spearheaded the project lifecycle management, formulating a comprehensive problem statement and architecting critical system components to ensure seamless alignment between technical implementation and project objectives.
- Created comprehensive travel domain ontology using Protégé, integrating technical aspects with user-centric design.

Sahana - Smart Event and Meetup Platform | *FastAPI, React.js, Firebase, GCP* — July 2024 - Present

- Developing a full-stack event management platform with React.js, FastAPI, and Firestore, featuring an interactive UI with Material-UI and Google Maps integration for location-based recommendations.
- Implemented OAuth authentication (Google SSO) and JWT-based login, along with a dynamic RSVP system using Firebase Cloud Messaging (FCM) for event notifications.
- Deployed backend on Google Cloud Platform (GCP) using Cloud Run for a scalable, serverless architecture and set up a CI/CD pipeline with GitHub Actions for automated deployment.